The Three Gap Theorem : Specification and Proof in Coq

Micaela Mayero

▶ **To cite this version:**

Micaela Mayero. The Three Gap Theorem : Specification and Proof in Coq. [Research Report] RR-3848, INRIA. 1999. inria-00072808

HAL Id: inria-00072808

https://inria.hal.science/inria-00072808

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

The Three Gap Theorem: Specification and Proof in Coq

Micaela Mayero

No 3848

Décembre 1999

THÈME 2

ISSN 0249-6399 ISRN INRIA/RR--3848--FR+ENG

The Three Gap Theorem: Specification and Proof in Coq

Micaela Mayero

Thème 2 — Génie logiciel
et calcul symbolique
Projet Coq

Rapport de recherche n ° 3848 — Décembre 1999 — 30 pages

Abstract: We present a specification and a proof in Coq of the three gap theorem, or initially Steinhaus conjecture whose result is the following: let N be the distribution of points placed consecutively around a circle by an angle of α; then the points partition the circle into gaps of at most three different lengths.
We start by making an axiomatization of the real numbers in Coq in order to use them in the development. Thereafter, we define all the mathematical tools necessary and some lemmas used in the proof. Finally we state the theorem.

Key-words: Coq, real numbers, specification, proof, theorem, three gap theorem

(Résumé : tsvp)

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : 01 39 63 55 11 - International : +33 1 39 63 55 11
Télécopie : (33) 01 39 63 53 30 - International : +33 1 39 63 53 30

Le théorème des trois intervalles: spécification et preuve en Coq

Résumé : Nous présentons une spécification et une preuve en Coq du théorème des trois intervalles, ou initialement conjecture de Steinhaus dont le résultat est le suivant: soient N points placés consécutivement autour d'un cercle modulo un angle α; les N points partitionnent le cercle en au plus trois longueurs différentes d'intervalle.
Pour ce faire, nous commencerons par faire une axiomatisation des nombres réels dans Coq afin de pouvoir les utiliser au cours du développement. Nous définirons par la suite tous les outils mathématiques nécessaires ainsi que certains lemmes intermédiaires utilisés dans la preuve. Enfin nous énoncerons le théorème.

Mots-clé : Coq, nombres réels, spécification, preuve, théorème, théorème des trois intervalles

Contents

Introduction **4**

1 The real numbers **5**
1.1 The choice for the definition of the real numbers in Coq 6
1.2 The axiomatization of the real numbers 6
1.3 Some lemmas 9
1.3.1 The total order 9
1.3.2 The addition 10
1.3.3 The multiplication 10
1.3.4 The opposite and the subtraction 11
1.3.5 The "less than" relation 11
1.3.6 The "greater than" relation 13
1.3.7 The injection of $\mathbb{N}$ in $\mathbb{R}$ 15
1.3.8 The injection of $\mathbb{Z}$ in $\mathbb{R}$ 15
1.4 The integer part and the fractional part 16
1.4.1 Definition of the integer part 16
1.4.2 Definition of the fractional part 16
1.4.3 Some elementary properties 16
1.4.4 The subtraction of integer parts 17
1.4.5 The subtraction of fractional parts 17
1.4.6 The addition of integer parts 17
1.4.7 The addition of fractional parts 18

2 The three gap theorem **19**
2.1 Statement of the theorem 19
2.2 Proof of the theorem 22
2.2.1 Stage of the proof 22
2.2.2 Statement of the theorem 26

Conclusion **28**

Introduction

A statement of the three gap theorem can be the following: let N be a distribution of points placed consecutively around a circle by an angle of α, then the points partition the circle into gaps of at most three different lengths. Originally, this theorem was a conjecture by H. Steinhaus. Subsequently, several statements and proofs have been given [SOS57] [SOS58] [SWI58] [SUR58] [HAL65] [SLA67] [RAV88]. The process of demonstration we present in this report is based on Tony van Ravenstein's proof [RAV88].

This kind of demostration, which involves geometrical intuition, is a real challenge for proof assistance systems. That is what motivated our work. Therefore, the interest of such an approach is to understand, by means of an example, if the Coq system allows us to prove a theorem coming from pure mathematics.

This development has been automatically verified with version 6.3 of the Coq proof assistance system [BB+97] developed in the INRIA Rocquencourt and in the LRI Orsay. Coq and this contribution are available at
`http://pauillac.inria.fr/coq/assis-eng.html`.

As we will see throughout this report, the real numbers ($\mathbb{R}$) together with many of their properties play an important part both in the formalization and in the proof. Therefore, the first part of this report deals with formalizing the real numbers, as well as some properties of the integer and fractional parts for instance, in the Coq proof assistance system, making clear the choices taken for the mathematics and for Coq. In the second part, we present the tools required for the formalization of the theorem, as also the development from its proof to its statement.

This development allowed us to clarify some points of the proof and has led to a more detailed proof of the theorem, presented in [MAY98].

Chapter 1

The real numbers

In this chapter, we present the formalization of the real numbers used within the framework of the theorem.
Several choices are possible:

- The choice between a constructive analysis and a classical analysis.
  In its "usual" formalization, the three gap theorem has a classical proof. Therefore, we can raise the question of the existence of a constructive proof of the three gap theorem. We could probably give an intuitionistic proof for each of the two cases, according the whether α is rational or irrational because we know exactly the length of the gaps between two points of the circle. But, the two cases cannot be treated in a uniform way. Thus in our proof it should be supposed that α is rational or not and so far we do not see, how to avoid this distinction.
  Moreover, it is interesting to notice that the theorem shown is stronger than that which was stated initially. Indeed, not only do we show that there are at most three different lengths of gaps, but we can also give their value and their place on the circle. This modified statement is due to [RAV88].

- The choice between a construction and an axiomatization of the real numbers.
  Two possibilities exist to describe the real numbers: we can construct the reals or axiomatize them. We chose an axiomatical development for reasons of simplicity and rapidity. We can refer to [LAN71] and [LAN51] for constructions from Cauchy's sequences or Dedikind's cuts.
  Thereafter, we deal with the axiomatization itself.

1.1 The choice for the definition of the real numbers in Coq

Therefore, we use a classical axiomatization of the real numbers. This axiomatization allows us to build functions of the real numbers into the real numbers which are not continuous, and therefore not recursive. For this reason, we place the type $\mathbb{R}$ of the real numbers into the sort `Type`. Indeed, if we term |A| the interpretation of a type A, then |A→B| is the set of the computable functions of |A| into |B| when A and B are of type `Set`, whereas it is the set of all the functions of |A| into |B| when A and B are of type `Type`.

So, we must define into `Type` the equivalents of the usual disjonctions and quantification (`sumbool`, `sumor` and `sig` connectors) defined in `Set`.

```
Inductive sumboolT [A,B:Prop]:Type:=
   leftT : A->(sumboolT A B)
  |rightT: B->(sumboolT A B).

Inductive sumorT [A:Type;B:Prop]:Type:=
   inleftT : A->(sumorT A B)
  |inrightT: B->(sumorT A B).

Inductive sigT [A:Set;P:A->Prop]:Type:=
   existT: (x:A)(P x)->(sigT A P).
```

1.2 The axiomatization of the real numbers

Most properties of the real numbers (commutative field, order, the Archimedian axiom) are first order properties. On the other hand, the completeness property is a second order property, as it requieres to quantify on the sets of real numbers. Instead of this axiom, we can put an infinity of first order axioms, according to which any odd degree has a root in $\mathbb{R}$. Hence, we get the "real closed field" notion.

We thus chose axiomatization at the *second order* based on the fact that $\mathbb{R}$ is a **commutative ordered Archimedian and complete field**. For these notions, we based our work on [DIE68] and [HAR96]. It should be pointed out that this choice also constrains the use of classical logic, seeing that an intuitionist reading of the total order involves the decidability of the equality of the real numbers, which obviously, is not the case.

We begin by defining the basic constructors:

- The type of real numbers.

```
Parameter R:Type.
```

- Field constructors.

```
Parameter R0:R.
Parameter R1:R.
Parameter Rplus:R->R->R.
Parameter Rmult:R->R->R.
Parameter Ropp:R->R.
Parameter Rinv:R->R.
```

- Order constructor.

```
Parameter Rlt:R->R->Prop.
```

- The constructor of the Archimedes axiom.

```
Parameter up:R->Z.
```

The order used in the axiomatization is the relation $<$. For greater clarity of the axioms, we define constructors $>$, $\leq$ and $\geq$ as follows:

```
Definition Rgt:R->R->Prop:=[r1,r2:R](Rlt r2 r1).
Definition Rle:R->R->Prop:=[r1,r2:R]((Rlt r1 r2)\/(r1==r2)).
Definition Rge:R->R->Prop:=[r1,r2:R]((Rgt r1 r2)\/(r1==r2)).
```

In the same way, we define the function minus:

```
Definition Rminus:R->R->R:=[r1,r2:R](Rplus r1 (Ropp r2)).
```

The axioms of the commutative field are the following:

- Properties of addition.

```
Axiom Rplus_sym:(r1,r2:R)((Rplus r1 r2)==(Rplus r2 r1)).

Axiom Rplus_assoc:(r1,r2,r3:R)
  ((Rplus (Rplus r1 r2) r3)==(Rplus r1 (Rplus r2 r3))).

Axiom Rplus_Ropp_r:(r:R)((Rplus r (Ropp r))==R0).

Axiom Rplus_ne:(r:R)(((Rplus r R0)==r)/\((Rplus R0 r)==r)).
```

- Properties of multiplication.

```
Axiom Rmult_sym:(r1,r2:R)((Rmult r1 r2)==(Rmult r2 r1)).

Axiom Rmult_assoc:(r1,r2,r3:R)
  ((Rmult (Rmult r1 r2) r3)==(Rmult r1 (Rmult r2 r3))).

Axiom Rinv_l:(r:R)(~(r==R0))->((Rmult (Rinv r) r)==R1).

Axiom Rmult_ne:(r:R)(((Rmult r R1)==r)/\((Rmult R1 r)==r)).

Axiom R1_neq_R0:(~R1==R0).
```

- The distributivity.

```
Axiom Rmult_Rplus_distr:(r1,r2,r3:R)
  ((Rmult r1 (Rplus r2 r3))==(Rplus (Rmult r1 r2) (Rmult r1 r3))).
```

The order axioms:

- Total order in Prop.

```
Axiom total_order:(r1,r2:R)((Rlt r1 r2)\/(r1==r2)\/(Rgt r1 r2)).
```

- Total order in Type.

```
Axiom total_order_Rle:(r1,r2:R)(sumboolT (Rle r1 r2) ~(Rle r1 r2)).
```

- Lower.

```
Axiom Rlt_antisym:(r1,r2:R)(Rlt r1 r2)->~(Rlt r2 r1).

Axiom Rlt_trans:(r1,r2,r3:R)
  (Rlt r1 r2)->(Rlt r2 r3)->(Rlt r1 r3).
```

- Compatibility of addition and multiplication.

```
Axiom Rlt_compatibility:(r,r1,r2:R)(Rlt r1 r2)->
      (Rlt (Rplus r r1) (Rplus r r2)).

Axiom Rlt_monotony:(r,r1,r2:R)(Rlt R0 r)->(Rlt r1 r2)->
      (Rlt (Rmult r r1) (Rmult r r2)).
```

The Archimedes axiom.
We begin by defining the injections of $\mathbb{N}$ in $\mathbb{R}$ and of $\mathbb{Z}$ in $\mathbb{R}$.

```
Fixpoint INR [n:nat]:R:=(Cases n of
                          O      => R0
                         |(S O)  => R1
                         |(S n)  => (Rplus (INR n) R1)
                        end).

Definition IZR:Z->R:=[z:Z](Cases z of
                         ZERO      => R0
                        |(POS n) => (INR (convert n))
                        |(NEG n) => (Ropp (INR (convert n)))
                           end).
```

The variable **(up r)** represents the integer immediately above the real number r. Our formulation of Archimedian axiom is stronger than the one usually stated (the existence of an integer above any real) but it is equivalent, since any ordered and minored set of integers has a least element.

```
Axiom archimed:(r:R)(Rgt (IZR (up r)) r)/\
                    (Rle (Rminus (IZR (up r)) r) R1).
```

The completeness axiom.

```
Definition is_upper_bound:=[E:R->Prop][m:R](x:R)(E x)->(Rle x m).

Definition bound:=[E:R->Prop](ExT [m:R](is_upper_bound E m)).

Definition is_lub:=[E:R->Prop][m:R]
    (is_upper_bound E m)/\(b:R)(is_upper_bound E b)->(Rlt m b).

Axiom complet:(E:R->Prop)((bound E)->(ExT [m:R](is_lub E m))).
```

From this axiomatization, we have proved some elementary properties.

1.3 Some lemmas

1.3.1 The total order

```
Lemma Req_EM:(r1,r2:R)(r1==r2)\/(~r1==r2).

Lemma not_Req:(r1,r2:R)(~(r1==r2))->((Rlt r1 r2)\/(Rgt r1 r2)).

Lemma imp_not_Req:(r1,r2:R)((Rlt r1 r2)\/(Rgt r1 r2))->(~(r1==r2)).
```

1.3.2 The addition

```
Lemma Rplus_Ropp_l:(r:R)((Rplus (Ropp r) r)==R0).

Lemma Rplus_Ropp:(x,y:R)((Rplus x y)==R0)->(y==(Ropp x)).

Lemma Rplus_plus_r:(r,r1,r2:R)(r1==r2)->((Rplus r r1)==(Rplus r r2)).

Lemma r_Rplus_plus:(r,r1,r2:R)((Rplus r r1)==(Rplus r r2))->(r1==r2).

Lemma Rplus_ne_i:(r,b:R)((Rplus r b)==r)->(b==R0).
```

1.3.3 The multiplication

```
Lemma Rinv_r:(r:R)(~(r==R0))->((Rmult r (Rinv r))==R1).

Lemma Rmult_0r:(r:R)((Rmult r R0)==R0).

Lemma Rmult_0l:(r:R)((Rmult R0 r)==R0).

Lemma Rmult_mult_r:(r,r1,r2:R)(r1==r2)->((Rmult r r1)==(Rmult r r2)).

Lemma r_Rmult_mult:(r,r1,r2:R)((Rmult r r1)==(Rmult r r2))->
  (~(r==R0))->(r1==r2).

Lemma without_div_0d:(r1,r2:R)(Rmult r1 r2)==R0 -> r1==R0 \/ r2==R0.

Lemma without_div_0i1:(r1,r2:R) r1==R0 -> (Rmult r1 r2)==R0.

Lemma without_div_0i2:(r1,r2:R) r2==R0 -> (Rmult r1 r2)==R0.

Lemma without_div_0i:(r1,r2:R) (r1==R0)\/(r2==R0) -> (Rmult r1 r2)==R0.

Lemma without_div_0_contr:(r1,r2:R)
   ~(Rmult r1 r2)==R0 -> ~r1==R0 /\ ~r2==R0.

Definition Rsqr:R->R:=[r:R](Rmult r r).

Lemma Rsqr_0:((Rsqr R0)==R0).

Lemma Rsqr_r_R0:(r:R)(Rsqr r)==R0->r==R0.
```

1.3.4 The opposite and the subtraction

```
Lemma Ropp_Ropp:(r:R)((Ropp (Ropp r))==r).

Lemma Ropp_0:((Ropp R0)==R0).

Lemma Ropp_distr1:(r1,r2:R)
 ((Ropp (Rplus r1 r2))==(Rplus (Ropp r1) (Ropp r2))).

Lemma Ropp_distr2:(r1,r2:R)((Ropp (Rminus r1 r2))==(Rminus r2 r1)).

Lemma eq_Rminus:(r1,r2:R)(r1==r2)->((Rminus r1 r2)==R0).

Lemma Rminus_eq:(r1,r2:R)(Rminus r1 r2)==R0 -> r1==r2.

Lemma eq_Ropp:(r1,r2:R)(r1==r2)->((Ropp r1)==(Ropp r2)).

Lemma eq_Ropp0:(r:R)(r==R0)->((Ropp r)==R0).

Lemma minus_R0:(r:R)(Rminus r R0)==r.

Lemma Ropp_mul1:(r1,r2:R)(Rmult (Ropp r1) r2)==(Ropp (Rmult r1 r2)).

Lemma Ropp_mul2:(r1,r2:R)(Rmult (Ropp r1) (Ropp r2))==(Rmult r1 r2).

Lemma Rinv_Rmult:(r1,r2:R)(~r1==R0)->(~r2==R0)->
  (Rinv (Rmult r1 r2))==(Rmult (Rinv r1) (Rinv r2)).

Lemma Rinv_Rmult_simpl:(a,b,c:R)(~a==R0)->
      (Rmult (Rmult a (Rinv b))(Rmult c (Rinv a)))==
      (Rmult c (Rinv b)).
```

1.3.5 The "less than" relation

```
Lemma Rlt_antirefl:(r:R)~(Rlt r r).

Lemma not_Rle:(r1,r2:R)~(Rle r1 r2)->(Rgt r1 r2).

Lemma Rle_not:(r1,r2:R)(Rlt r2 r1)->~(Rle r1 r2).

Lemma Rle_le_eq:(r1,r2:R)(Rle r1 r2)/\(Rle r2 r1)<->(r1==r2).

Lemma Rlt_le:(r1,r2:R)(Rlt r1 r2)->(Rle r1 r2).
```

```
Lemma eq_Rle:(r1,r2:R)(r1==r2)->(Rle r1 r2).

Lemma Rle_trans:(r1,r2,r3:R)
  (Rle r1 r2)->(Rle r2 r3)->(Rle r1 r3).

Lemma Rle_lt_trans:(r1,r2,r3:R)
  (Rle r1 r2)->(Rlt r2 r3)->(Rlt r1 r3).

Lemma Rlt_le_trans:(r1,r2,r3:R)
  (Rlt r1 r2)->(Rle r2 r3)->(Rlt r1 r3).

Lemma Rlt_anti_compatibility:
    (r,r1,r2:R)(Rlt (Rplus r r1) (Rplus r r2))->(Rlt r1 r2).

Lemma Rle_compatibility:(r,r1,r2:R)(Rle r1 r2)->
      (Rle (Rplus r r1) (Rplus r r2)).

Lemma Rle_anti_compatibility:
   (r,r1,r2:R)(Rle (Rplus r r1) (Rplus r r2))->(Rle r1 r2).

Lemma Rgt_Ropp:(r1,r2:R)
 (Rgt r1 r2)->(Rlt (Ropp r1) (Ropp r2)).

Lemma Rlt_Ropp:(r1,r2:R)
 (Rlt r1 r2)->(Rgt (Ropp r1) (Ropp r2)).

Lemma Rlt_anti_monotony:(r,r1,r2:R)(Rlt r R0)->(Rlt r1 r2)->
      (Rgt (Rmult r r1) (Rmult r r2)).

Lemma Rlt_minus:(r1,r2:R)(Rlt r1 r2)->(Rlt (Rminus r1 r2) R0).

Lemma Rle_minus:(r1,r2:R)(Rle r1 r2)->(Rle (Rminus r1 r2) R0).

Lemma Rminus_lt:(r1,r2:R)(Rlt (Rminus r1 r2) R0)->(Rlt r1 r2).

Lemma Rminus_le:(r1,r2:R)(Rle (Rminus r1 r2) R0)->(Rle r1 r2).

Lemma sum_inequa_Rle_lt:(a,x,b,c,y,d:R)(Rle a x)->(Rlt x b)->
           (Rlt c y)->(Rle y d)->
        (Rlt (Rplus a c) (Rplus x y))/\(Rlt (Rplus x y) (Rplus b d)).
```

```
Lemma Rplus_lt:(r1,r2,r3,r4:R)(Rlt r1 r2)->(Rlt r3 r4)->
 (Rlt (Rplus r1 r3) (Rplus r2 r4)).

Lemma Rmult_lt:(r1,r2,r3,r4:R)(Rgt r3 R0)->(Rgt r2 R0)->
  (Rlt r1 r2)->(Rlt r3 r4)->(Rlt (Rmult r1 r3) (Rmult r2 r4)).

Lemma inser_trans_R:(n,m,p,q:R)(Rle n m)/\(Rlt m p)->
      (sumboolT ((Rle n m)/\(Rlt m q)) ((Rle q m)/\(Rlt m p))).

Lemma tech_Rplus:(r,s:R)(Rle R0 r)->(Rlt R0 s)->~(Rplus r s)==R0.

Lemma pos_Rsqr:(r:R)(Rle R0 (Rsqr r)).

Lemma pos_Rsqr1:(r:R)(~(r==R0))->(Rlt R0 (Rsqr r)).

Lemma Rlt_R0_R1:(Rlt R0 R1).

Lemma Rlt_Rinv:(r:R)(Rlt R0 r)->(Rlt R0 (Rinv r)).
```

1.3.6 The "greater than" relation

```
Lemma Rge_ge_eq:(r1,r2:R)(Rge r1 r2)->(Rge r2 r1)->r1==r2.

Lemma Rlt_not_ge:(r1,r2:R)~(Rlt r1 r2)->(Rge r1 r2).

Lemma Rgt_not_le:(r1,r2:R)~(Rgt r1 r2)->(Rle r1 r2).

Lemma Rgt_ge:(r1,r2:R)(Rgt r1 r2)->(Rge r1 r2).

Lemma Rlt_sym:(r1,r2:R)
 (((Rlt r1 r2)->(Rgt r2 r1))/\((Rgt r2 r1)->(Rlt r1 r2))).

Lemma Rle_sym1:(r1,r2:R)(Rle r1 r2)->(Rge r2 r1).

Lemma Rle_sym2:(r1,r2:R)(Rge r2 r1)->(Rle r1 r2).

Lemma Rle_sym:(r1,r2:R)
 (((Rle r1 r2)->(Rge r2 r1))/\((Rge r2 r1)->(Rle r1 r2))).

Lemma Rge_gt_trans:(r1,r2,r3:R)
 (Rge r1 r2)->(Rgt r2 r3)->(Rgt r1 r3).

Lemma Rgt_ge_trans:(r1,r2,r3:R)
```

```
 (Rgt r1 r2)->(Rge r2 r3)->(Rgt r1 r3).

Lemma Rgt_trans:(r1,r2,r3:R)
 (Rgt r1 r2)->(Rgt r2 r3)->(Rgt r1 r3).

Lemma Rge_trans:(r1,r2,r3:R)
 (Rge r1 r2)->(Rge r2 r3)->(Rge r1 r3).

Lemma eq_Rge:(r1,r2:R)(r1==r2)->(Rge r1 r2).

Lemma Rle_Ropp:(r1,r2:R)
 (Rle r1 r2)->(Rge (Ropp r1) (Ropp r2)).

Lemma Rgt_Ropp0:(r:R)(Rgt r R0)->(Rlt (Ropp r) R0).

Lemma Rlt_Ropp0:(r:R)(Rlt r R0)->(Rgt (Ropp r) R0).

Lemma Rlt_r_plus_R1:(r:R)(Rle R0 r)->(Rlt R0 (Rplus r R1)).

Lemma tech_Rgt_minus:(r1,r2:R)(Rlt R0 r2)->(Rgt r1 (Rminus r1 r2)).

Lemma Rgt_plus_plus_r:(r,r1,r2:R)(Rgt r1 r2)->
      (Rgt (Rplus r r1) (Rplus r r2)).

Lemma Rgt_r_plus_plus:(r,r1,r2:R)(Rgt (Rplus r r1) (Rplus r r2))->
                             (Rgt r1 r2).

Lemma Rge_plus_plus_r:(r,r1,r2:R)(Rge r1 r2)->
      (Rge (Rplus r r1) (Rplus r r2)).

Lemma Rge_r_plus_plus:(r,r1,r2:R)(Rge (Rplus r r1) (Rplus r r2))->
                             (Rge r1 r2).

Lemma Rgt_minus:(r1,r2:R)(Rgt r1 r2)->(Rgt (Rminus r1 r2) R0).

Lemma Rge_minus:(r1,r2:R)(Rge r1 r2)->(Rge (Rminus r1 r2) R0).

Lemma minus_Rge:(r1,r2:R)(Rge (Rminus r1 r2) R0)->(Rge r1 r2).

Lemma Rmult_gt:(r1,r2:R)(Rgt r1 R0)->(Rgt r2 R0)->
                        (Rgt (Rmult r1 r2) R0).
```

```
Lemma Rplus_eq_R0:(a,b:R)(Rle R0 a)->(Rle R0 b)->(Rplus a b)==R0->
                          (a==R0)/\(b==R0).

Lemma Rplus_Rsr_eq_R0:(a,b:R)(Rplus (Rsqr a) (Rsqr b))==R0->
                          (a==R0)/\(b==R0).
```

1.3.7 The injection of $\mathbb{N}$ in $\mathbb{R}$

```
Lemma S_INR:(n:nat)(INR (S n))==(Rplus (INR n) R1).

Lemma S_O_plus_INR:(n:nat)
    (INR (plus (S O) n))==(Rplus (INR (S O)) (INR n)).

Lemma plus_INR:(n,m:nat)(INR (plus n m))==(Rplus (INR n) (INR m)).

Lemma minus_INR:(n,m:nat)(le m n)->
   (INR (minus n m))==(Rminus (INR n) (INR m)).

Lemma INR_le:(n:nat)(Rle R0 (INR n)).

Lemma not_INR_O:(n:nat)~(INR n)==R0->~n=O.

Lemma not_O_INR:(n:nat)~n=O->~(INR n)==R0.
```

1.3.8 The injection of $\mathbb{Z}$ in $\mathbb{R}$

```
Definition INZ:=inject_nat.

Lemma INR_IZR_INZ:(n:nat)(INR n)==(IZR (INZ n)).

Lemma plus_IZR:(z,t:Z)(IZR (Zplus z t))==(Rplus (IZR z) (IZR t)).

Lemma Ropp_Ropp_IZR:(z:Z)(IZR (`-z`))==(Ropp (IZR z)).

Lemma Z_R_minus:(z1,z2:Z)
  (Rminus (IZR z1) (IZR z2))==(IZR (Zminus z1 z2)).

Lemma lt_O_IZR:(z:Z)(Rlt R0 (IZR z))->(Zlt ZERO z).

Lemma lt_IZR:(z1,z2:Z)(Rlt (IZR z1) (IZR z2))->(`z1<z2`).

Lemma eq_IZR_R0:(z:Z)(IZR z)==R0->`z=0`.
```

```
Lemma eq_IZR:(z1,z2:Z)(IZR z1)==(IZR z2)->z1=z2.

Lemma le_IZR_R1:(z:Z)(Rle (IZR z) R1)->(Zle z `1`).

Lemma single_z_r_R1:(r:R)(z,x:Z)(Rlt r (IZR z))->
                                (Rle (IZR z) (Rplus r R1))->
             (Rlt r (IZR x))->(Rle (IZR x) (Rplus r R1))->z=x.

Lemma tech_single_z_r_R1:(r:R)(z:Z)(Rlt r (IZR z))->
                                     (Rle (IZR z) (Rplus r R1))->
 (Ex [s:Z] (~s=z/\(Rlt r (IZR s))/\(Rle (IZR s) (Rplus r R1))))->False.
```

1.4 The integer part and the fractional part

1.4.1 Definition of the integer part

```
Definition Int_part:R->Z:=[r:R](`(up r)-1`).
```

1.4.2 Definition of the fractional part

```
Definition frac_part:R->R:=[r:R](Rminus r (IZR (Int_part r))).
```

Now, we state some elementary properties on the fractional part and the integer part. They will be used to prove the properties about the subtraction and the addition of two integer parts or two fractional parts.

1.4.3 Some elementary properties

```
Lemma tech_up:(r:R)(z:Z)(Rlt r (IZR z))->(Rle (IZR z) (Rplus r R1))->
  z=(up r).

Lemma up_tech:(r:R)(z:Z)(Rle (IZR z) r)->(Rlt r (IZR `z+1`))->
  `z+1`=(up r).

Lemma fp_R0:(frac_part R0)==R0.

Lemma for_base_fp:(r:R)(Rgt (Rminus (IZR (up r)) r) R0)/\
                       (Rle (Rminus (IZR (up r)) r) R1).

Lemma base_fp:(r:R)(Rge (frac_part r) R0)/\(Rlt (frac_part r) R1).

Lemma base_Int_part:(r:R)(Rle (IZR (Int_part r)) r)/\
                    (Rgt (Rminus (IZR (Int_part r)) r) (Ropp R1)).
```

```
Lemma fp_nat:(r:R)(frac_part r)==R0->(Ex [c:Z](r==(IZR c))).

Lemma R0_fp_O:(r:R)~R0==(frac_part r)->~R0==r.
```

We are particulary interested in four properties: those concerning the addition and the subtraction of two fractional parts. To prove them, we use the four similar properties concerning the integer part.

- if $\{r1\} + \{r2\} \geq 1$ then $\{r1 + r2\} = \{r1\} + \{r2\} - 1$
- if $\{r1\} + \{r2\} < 1$ then $\{r1 + r2\} = \{r1\} + \{r2\}$
- if $\{r1\} \geq \{r2\}$ then $\{r1 - r2\} = \{r1\} - \{r2\}$
- if $\{r1\} < \{r2\}$ then $\{r1 - r2\} = \{r1\} - \{r2\} + 1$

1.4.4 The subtraction of integer parts

```
Lemma Rminus_Int_part1:(r1,r2:R)(Rge (frac_part r1) (frac_part r2))->
  (Int_part (Rminus r1 r2))=(Zminus (Int_part r1) (Int_part r2)).

Lemma Rminus_Int_part2:(r1,r2:R)(Rlt (frac_part r1) (frac_part r2))->
  (Int_part (Rminus r1 r2))=
  (Zminus (Zminus (Int_part r1) (Int_part r2)) `1`).
```

1.4.5 The subtraction of fractional parts

```
Lemma Rminus_fp1:(r1,r2:R)(Rge (frac_part r1) (frac_part r2))->
  (frac_part (Rminus r1 r2))==(Rminus (frac_part r1) (frac_part r2)).

Lemma Rminus_fp2:(r1,r2:R)(Rlt (frac_part r1) (frac_part r2))->
  (frac_part (Rminus r1 r2))==
  (Rplus (Rminus (frac_part r1) (frac_part r2)) R1).
```

1.4.6 The addition of integer parts

```
Lemma plus_Int_part1:(r1,r2:R)
          (Rge (Rplus (frac_part r1) (frac_part r2)) R1)->
  (Int_part (Rplus r1 r2))=
  (Zplus (Zplus (Int_part r1) (Int_part r2)) `1`).

Lemma plus_Int_part2:(r1,r2:R)
          (Rlt (Rplus (frac_part r1) (frac_part r2)) R1)->
  (Int_part (Rplus r1 r2))=(Zplus (Int_part r1) (Int_part r2)).
```

1.4.7 The addition of fractional parts

```
Lemma plus_frac_part1:(r1,r2:R)
  (Rge (Rplus (frac_part r1) (frac_part r2)) R1)->
               (frac_part (Rplus r1 r2))==
               (Rminus (Rplus (frac_part r1) (frac_part r2)) R1).

Lemma plus_frac_part2:(r1,r2:R)
  (Rlt (Rplus (frac_part r1) (frac_part r2)) R1)->
(frac_part (Rplus r1 r2))==(Rplus (frac_part r1) (frac_part r2)).
```

Chapter 2

The three gap theorem

2.1 Statement of the theorem

In order to understand more quickly the notations and requirements, our reference will be, from now onwards, figure 2.1.

The indispensable first definitions for the specification of the theorem are: the definition of **first** - the first point to the right of zero on the circle -, **last** - the first point to the left of zero on the circle -, and **after** - the successor of a point. As already said, N is the number of points on the circle.

Let us start with the angle α. The unity for angle is by revolutions, therefore α is between 0 and 1, and it cannot be 0.

```
Variable alpha:R.
Hypothesis prop_alpha:((Rlt R0 alpha)/\(Rlt alpha R1)).
```

The function **part_frac_n_alpha** associates with every integer n the distance between the first point on the circle - written 0- and the n^{th} point.

```
Definition frac_part_n_alpha:nat->R:=[n:nat]
  (frac_part (Rmult (INR n) alpha)).
```

The function **ordre_total** defines a new order for the points of the circle according to their distance to 0.

```
Definition ordre_total:=[n,m:nat]((Rlt R0 alpha)/\(Rlt alpha R1))->
  (Rle (frac_part_n_alpha n) (frac_part_n_alpha m)).
```

The following lemmas allows us to define **first** and **last**:

- we first prove that such points exist

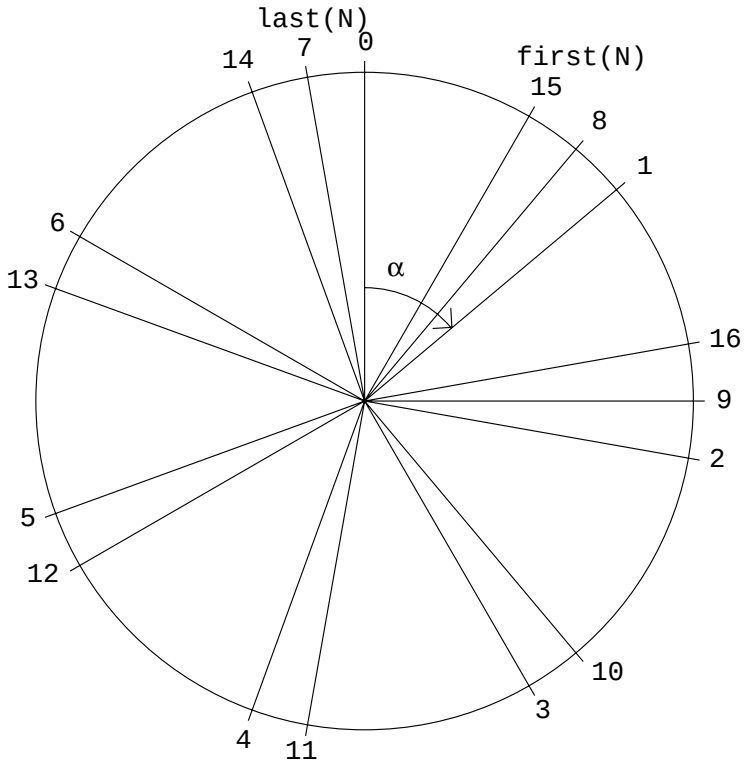

Figure 2.1: The three gap theorem

```
Lemma N_classic:(N:nat){N=O}+{N=(S O)}+{(ge N (S (S O)))}.

Lemma tech_total_order:(n:nat)(ordre_total n n).

Lemma exist_first:(N:nat)(ge N (S (S O)))->
                      (sigT nat [n:nat]((lt O n)/\(lt n N)/\
                             (m:nat)(lt O m)/\(lt m N)
                             ->(ordre_total n m))).

Lemma  exist_last:(N:nat)(ge N (S (S O)))->
                      (sigT nat [n:nat] ((lt O n)/\(lt n N)/\
                             (m:nat)(lt O m)/\(lt m N)
                            ->(ordre_total m n))).
```

- then we name the object

```
Definition first:=[N:nat](Cases (N_classic N) of
  (inright p)=>(<nat>Cases (exist_first N p) of (existT a b)=>a end)
 |   _       => O
                        end).

Definition last:=[N:nat](Cases (N_classic N) of
  (inright p)=> (<nat>Cases (exist_last N p) of (existT a b)=>a end)
 |   _       => O
                        end).
```

We define the function **after** in the same way.

```
Lemma exist_after_M:(M:R)(Rle R0 M)->(Rlt M R1)->(N:nat)
                      (sumorT (sigT nat [I:nat]((lt O I)/\(lt I N)/\
                           (Rlt M (frac_part_n_alpha I))/\
                           (m:nat)(le O m)->(lt m N)->
                           (Rgt (frac_part_n_alpha m) M)->
                           (Rge (frac_part_n_alpha m)
                                (frac_part_n_alpha I))))
                      ((m:nat)(le O m)->(lt m N)->
                           (Rle R0 (frac_part_n_alpha m))/\
                           (Rle (frac_part_n_alpha m) M))).

Lemma P1:(n:nat)(Rle R0 (frac_part_n_alpha n)).
Lemma P2:(n:nat)(Rlt (frac_part_n_alpha n) R1).

Definition after:=[N,n:nat]
```

```
  (Cases (exist_after_M (frac_part_n_alpha n) (P1 n) (P2 n) N) of
      (inleftT p) => (<nat>Cases p of (existT a b)=>a end)
    | _         => 0
   end).
```

2.2 Proof of the theorem

2.2.1 Stage of the proof

First, we need to prove some simple properties on **first** and **last**. They are proved almost directly from their definition.

```
Lemma first_N:(N:nat)(ge N (S(S O)))->(lt (first N) N).

Lemma last_N:(N:nat)(ge N (S(S O)))->(lt (last N) N).

Lemma inter1:(N:nat)(ge N (S(S O)))->(lt (max (first N) (last N)) N).

Lemma first_N01:(N:nat)(le (first N) N).

Lemma last_N01:(N:nat)(le (last N) N).

Lemma first_0:(N:nat)(ge N (S(S O)))->(lt O (first N)).

Lemma last_0:(N:nat)(ge N (S(S O)))->(lt O (last N)).

Lemma first_n:(N,n:nat)(ge N (S(S O)))->(lt O n)->(lt n N)->
           ((Rlt R0 alpha)/\(Rlt alpha R1))->
           (Rle (frac_part_n_alpha (first N)) (frac_part_n_alpha n)).

Lemma last_n:(N,n:nat)(ge N (S(S O)))->(lt O n)->(lt n N)->
           ((Rlt R0 alpha)/\(Rlt alpha R1))->
           (Rle (frac_part_n_alpha n) (frac_part_n_alpha (last N))).

Lemma tech_first_last:(N,k:nat)(ge N (S(S O)))->(lt O k)->(lt k N)->
                      (ordre_total (first N) k)/\
                      (ordre_total k (last N)).

Lemma le_first_last:(N:nat)(ge N (S(S O)))->
                    (ordre_total (first N) (last N)).
```

We can observe that, so far , the specification is available for any value of integer N and for any real number α. Henceforth, we do not deal with the degenerated cases such as N=0

or N=1 because, in these cases, there is only one length of gap, and cases such as α rational ($\alpha = p/q$) because here there are only one or two lengths of gap, depending on whether N$\geq q$ or N$< q$. Therefore, we put:

```
Section theoreme.
Hypothesis alpha_irr:(n,p:Z)~(Rmult alpha (IZR p))==(IZR n).
Hypothesis prop_alpha:((Rlt R0 alpha)/\(Rlt alpha R1)).
Hypothesis prop_N:(N:nat)(ge N (S (S O))).
```

The three following lemmas are verified only for the α considered and they are used to prove the lemmas below.

```
Lemma tech_fp_alp_irr:(n,m:nat)
   (frac_part_n_alpha n)==(frac_part_n_alpha m)->n=m.

Lemma fp_first_R0:(N:nat)(Rgt (frac_part_n_alpha (first N)) R0).

Lemma contra_tech_fp_alp_irr:(n,m:nat)
   ~n=m->~(frac_part_n_alpha n)==(frac_part_n_alpha m).
```

The proof of the theorem is based on the idea that if we have (first N)+(last N) points on the circle, then there are only two lengths of gaps (see p.25). Then, the issue is to consider, initially, M=(first N)+(last N) points and to remove the M-N points in excess. So, we must compare the **first**, **last** and **after** for N and M. The main important results are the three lemmas:

- **first_eq_M_N**: if M=(first N)+(last N) then first(N)=first(M)
- **last_eq_M_N**: if M=(first N)+(last N) then last(N)=last(M)
- **le_N_M**: if M=(first N)+(last N) then N$\leq$ M

and so:

- if M=(first N)+(last N) then M=(first M)+(last M)

- General properties when we have M points on the circle.

```
Lemma contradiction1:(N,k:nat)(le N k)->
                              (lt k (plus (first N) (last N)))->
      ((Rlt (frac_part_n_alpha k) (frac_part_n_alpha (first N)))\/
      (Rgt (frac_part_n_alpha k) (frac_part_n_alpha (last N))))->
      False.

Lemma absurd1:(N,k:nat)(le N k)->(lt k (plus (first N) (last N)))->
      (Rge (frac_part_n_alpha k) (frac_part_n_alpha (first N)))/\
      (Rle (frac_part_n_alpha k) (frac_part_n_alpha (last N))).
```

- Let us define **first_eq_M_N** and **last_eq_M_N**.

```
Lemma absurd_first:(N,k:nat)
                   (lt O k)->(lt k (plus (first N) (last N)))->
      (Rle (frac_part_n_alpha (first N)) (frac_part_n_alpha k)).

Lemma absurd_last:(N,k:nat)
                   (lt O k)->(lt k (plus (first N) (last N)))->
      (Rle (frac_part_n_alpha k) (frac_part_n_alpha (last N))).

Lemma tech_first_aux:(N:nat)(a:nat)(lt O a)->(lt a N)->
     ((b:nat)(lt O b)->(lt b N)->
      (Rle (frac_part_n_alpha a) (frac_part_n_alpha b)))<->
       a=(first N).

Lemma eq_first_M_N:(N:nat)(M:nat)(M=(plus (first N) (last N)))->
                   ((b:nat)(lt O b)->(lt b M)->
   (Rle (frac_part_n_alpha (first N)) (frac_part_n_alpha b)))<->
   (first N)=(first M).

Lemma first_eq_M_N:(N:nat)(M:nat)(M=(plus (first N) (last N)))->
                   (first N)=(first M).

Lemma tech_last_aux:(N:nat)(a:nat)(lt O a)->(lt a N)->
     ((b:nat)(lt O b)->(lt b N)->
      (Rle (frac_part_n_alpha b) (frac_part_n_alpha a)))<->
       a=(last N).

Lemma eq_last_M_N:(N:nat)(M:nat)(M=(plus (first N) (last N)))->
                   ((b:nat)(lt O b)->(lt b M)->
   (Rle (frac_part_n_alpha b ) (frac_part_n_alpha (last N))))<->
   (last N)=(last M).

Lemma last_eq_M_N:(N:nat)(M:nat)(M=(plus (first N) (last N)))->
                   (last N)=(last M).
```

- We state here some similar properties about **after**, which will be used afterwards.

```
Lemma tech_after:(N:nat)(n,m:nat)(lt O n)->(lt n N)->
                                 (le O m)->(lt m N)->
                (Rlt (frac_part_n_alpha n) (frac_part_n_alpha m))->
   ((Ex [k:nat] (lt O k)/\(lt k N)/\
                 (Rlt (frac_part_n_alpha n) (frac_part_n_alpha k))/\
                 (Rlt (frac_part_n_alpha k) (frac_part_n_alpha m)))->
       False)->m=(after N n).
```

```
Lemma prop_after:(N,n,m:nat)(after N n)=m->
 (Ex [k:nat] (lt O k)/\(lt k N)/\
 (Rlt (frac_part_n_alpha n) (frac_part_n_alpha k))/\
 (Rlt (frac_part_n_alpha k) (frac_part_n_alpha m)))->False.

Lemma tech_after_lt:(N:nat)(n,m:nat)(lt O n)->(lt n N)->
                    (~m=O)->(after N n)=m->
      (Rlt (frac_part_n_alpha n) (frac_part_n_alpha m)).

Lemma after_last:(N:nat)(after N (last N))=O.
```

- To show the lemma **le_N_M**, we use, among others, the following property:

```
Lemma prop_M:(N,M:nat)(M=(plus (first N) (last N)))->
  (Rlt (frac_part_n_alpha M) (frac_part_n_alpha (first N)))\/
  (Rgt (frac_part_n_alpha M) (frac_part_n_alpha (last N))).

Lemma le_N_M:(N,M:nat)(M=(plus (first N) (last N)))->(le N M).
End theoreme.
```

The two lemmas below correspond to the study of the circle containing M points. We can see distinctly that there are, at most, only two different lengths of gap.

```
Section particular.
Hypothesis alpha_irr:(n,p:Z)~(Rmult alpha (IZR p))==(IZR n).
Hypothesis prop_alpha:((Rlt R0 alpha)/\(Rlt alpha R1)).
Hypothesis prop_N:(N:nat)(ge N (S (S O))).

Definition M:=[N:nat](plus (first N) (last N)).

Lemma inter31a:(N:nat)(n:nat)(lt O n)->(lt n (last (M N)))->
                     (after (M N) n)=(plus n (first (M N))).

Lemma inter31b:(N:nat)(n:nat)(le (last (M N)) n)->(lt n (M N))->
                     (after (M N) n)=(minus n (last (M N))).
```

In order to use those two intermediate lemmas, we must now show that for the points "really" on the circle, i.e. from 0 to N, the successors correspond. That is to say, for all points (except the M-N points which have not yet been removed, i.e. the M-N points which are not "really" on the circle), $after(N) = after(M)$. The lemmas **eq_after_M_N1** and **eq_after_M_N2** transcribe this property.

```
Lemma tech1:(N:nat)(n:nat)(le (minus N (first N)) n)->
```

```
                          (lt n (last N))->
     (Rlt (frac_part_n_alpha n)
          (frac_part_n_alpha (minus (plus n (first N)) (last N)))).

Lemma tech_suc_N:(N:nat)(n:nat)(lt O n)->(lt n N)->
                        (k:nat)(lt O k)->(lt k N)->
          (Rlt (frac_part_n_alpha n) (frac_part_n_alpha k))->
          ~(frac_part_n_alpha k)==(frac_part_n_alpha (after N n))->
          (Rlt (frac_part_n_alpha (after N n)) (frac_part_n_alpha k)).

Lemma tech_suc_M:(N:nat)(n:nat)(le (minus N (first N)) n)->
                                (lt n (last N))->
   (Ex [k:nat] (lt O k)/\(lt k N)/\
       (Rlt (frac_part_n_alpha n) (frac_part_n_alpha k))/\
       (Rlt (frac_part_n_alpha k)
        (frac_part_n_alpha (minus (plus n (first N)) (last N))))->
   False.

Lemma tech_suc_M1:(N:nat)(n:nat)(le (minus N (first N)) n)->
                                (lt n (last N))->
   (k:nat) ~((lt O k)/\(lt k N)/\
       (Rlt (frac_part_n_alpha n) (frac_part_n_alpha k))/\
       (Rlt (frac_part_n_alpha k)
        (frac_part_n_alpha (minus (plus n (first N)) (last N))))).

Lemma eq_after_M_N1:(N,n:nat)(lt O n)->(lt n (minus N (first N)))->
                  (after (M N) n)=(after N n).

Lemma eq_after_M_N2:(N,n:nat)(le (last N) n)->(lt n N)->
                  (after (M N) n)=(after N n).

End particular.
```

2.2.2 Statement of the theorem

The three following lemmas correspond to the three lengths of the gaps. We can see that all the points of the circle are taken into account, which means that the existence of a fourth length is impossible.

```
Section Three.
Hypothesis alpha_irr:(n,p:Z)~(Rmult alpha (IZR p))==(IZR n).
Hypothesis prop_alpha:((Rlt R0 alpha)/\(Rlt alpha R1)).
Hypothesis prop_N:(N:nat)(ge N (S (S O))).
```

```
Lemma three_gap1:(N:nat)(n:nat)
                 (lt O n)->(lt n (minus N (first N)))->
                 (after N n)=(plus n (first N)).

Lemma three_gap2:(N:nat)(n:nat)
                   (le (minus N (first N)) n)->(lt n (last N))->
                   (after N n)=(minus (plus n (first N)) (last N)).

Lemma three_gap3:(N:nat)(n:nat)(le (last N) n)->(lt n N)->
                   (after N n)=(minus n (last N)).

End Three.
```

Below, we give a more compact and readable statement of the theorem, and which is closer to the theorem presented.

```
Section gap.
Variable N,n:nat.
Hypothesis alpha_irr:(n,p:Z)~(Rmult alpha (IZR p))==(IZR n).
Hypothesis prop_alpha:((Rlt R0 alpha)/\(Rlt alpha R1)).
Hypothesis prop_N:(N:nat)(ge N (S (S O))).
Hypothesis Hn:(lt O n)/\(lt n N).
Definition succes:=(after N n).
Definition num1:=(plus n (first N)).
Definition num2:=(minus (plus n (first N)) (last N)).
Definition num3:=(minus n (last N)).

Theorem three_gap:
      (succes=num1)\/(succes=num2)\/(succes=num3).

End gap.
```

Conclusion

In spite of the difficulty of such a formalization and proof for a proof assistance system, the theorem has been completely formalized in Coq. The main difficulties are the transcription of the geometrical intuition and the manipulation of the real numbers.

On the other hand, throughout this report, we confirmed that the Coq proof assistant system allows us to work out some purely mathematical proofs. To do so, we observe that some effort towards formalization is indispensable and that the resulting specification is stricter this, which allows us to transcribe these proofs more exactly in pure mathematical language. For more details, see [MAY98]. Within this framework, we can highlight some notable facts, which were emphazised :

- The axiom of completeness of the real numbers is not used. Therefore, this theorem is true for any commutative ordered Archimedian field.
- We clearly identified the degenerated cases: N= 0, N= 1 and α is rational.
- The irrationality of α appears explicitly and it is this property which allows us to show that all the points of the circle are distinct.
- The geometrical intuitions, which consist in observing,for instance, that $first(N) = first(M)$, are completely proved.

Conversely, such theorems can allow us to realize the limits of a proof assistant system. In our specific case, for instance, although the statement of the theorem mainly involves the natural numbers, the proof involves, to a large extent, the real numbers, which are an important part of the formalization.

Bibliography

[BB+97] Barras Bruno and co. *The Coq Proof Assistant Reference Manual Version 6.1.* Technical Report 0203, INRIA, May 1997.

[DIE68] Dieudonné Jean *Eléments d'analyse. Vol. 1. Fondements de l'analyse moderne.* Gauthier-Villars Paris (1968)

[HAL65] Halton J.H. *The distribution of the sequence* $\{\eta\xi\}$ $(n = 0, 1, 2, ...)$. Proc. Cambridge Phil. Soc. 71 (1965), 665-670

[HAR96] Harrison John Robert. *Theorem Proving with the Real Numbers.* Technical Report number 408, University of Cambridge Computer Laboratory, December 1996.

[LAN51] Landau Edmund *Foundations of analysis. The arithmetic of whole, rational, irrational and complex numbers.* Chelsea Publishing Company (1951)

[LAN71] Lang Serge *Algebra.* Addison-Wesley Publishing Company (1971)

[MAY98] Mayero Micaela *The three gap theorem (Steinhaus conjecture).* To appear ftp://ftp.inria.fr/INRIA/Projects/coq/Micaela.Mayero/PS/three-gap.ps.tgz

[RAV88] Van Ravenstein Tony *The three gap theorem (Steinhaus conjecture).* J. Austral. Math. Soc. (Series A) 45 (1988), 360-370

[SLA67] Slater N.B. *Gap and steps for the sequence* $\eta\theta$ *mod 1.* Proc. Cambridge Phil. Soc. 73 (1967), 1115-1122

[SOS57] Sós V.T. *On the theory of diophantine approximations.* Acta Math. Acad. Sci. Hungar. 8 (1957), 461-472

[SOS58] Sós V.T. *On the distribution mod 1 of the sequence* $\eta\alpha$. Ann. Univ. Sci. Budapest. Eötvös Sect. Math. 1 (1958), 127-134

[SWI58] Świerckowski S. *On successive settings of an arc on the circumference of a circle.* Fund. Math. 48 (1958), 187-189

[SUR58] Surányi J. *Über dei Anordnung der Vielfachen einer reelen Zahl mod 1.* Ann. Univ. Sci. Budapest. Eötvös Sect. Math. 1 (1958), 107-111

[TAR69] Tarski Alfred *What is elementary geometry?* Oxford Readings in Philosophy. Oxford University Press (1969)

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399